

**1 D4 - TEKKOM B**



**STRUKTUR DATA**



Nama : Septian Bagus Jumantoro

Kelas : 1 – D4 Teknik Komputer B

NRP : 3221600039

Dosen : Dr Bima Sena Bayu Dewantara S.ST, M.T

Mata Kuliah : Praktikum Pemrogaman Dasar 2

Hari/Tgl. Praktikum : Rabu, 16 Maret 2022

* **Tugas 1**

**Source Code**

#include <iostream>

#include <string.h>

using namespace std;

struct Books{

    char title[50];

    char author[50];

    char subject[100];

    int book\_id;

};

int main( ){

    system("cls");

    struct Books Book1; *// Declare Book1 of type Book*

    struct Books Book2; *// Declare Book2 of type Book*

*// book 1 specification*

    strcpy(Book1.title, "Learn C++ Programming" );

    strcpy(Book1.author, "Chand Miyan" );

    strcpy(Book1.subject, "C++ Programming" );

    Book1. book\_id = 6495407;

*// book 2 specification*

    strcpy(Book2.title, "Telecom Billing" );

    strcpy(Book2.author, "Yakit Singha");

    strcpy(Book2.subject, "Telecom");

    Book2. book\_id = 6495700;

*// Print Book1 info*

    cout << "Book 1 title : " << Book1.title <<endl;

    cout << "Book 1 author : " << Book1.author <<endl;

    cout << "Book 1 subject : " << Book1.subject <<endl;

    cout << "Book 1 id : " << Book1.book\_id <<endl;

*// Print Book2 info*

    cout << "Book 2 title : " << Book2.title <<endl;

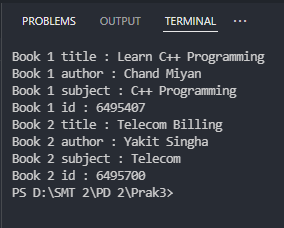
    cout << "Book 2 author : " << Book2.author <<endl;

    cout << "Book 2 subject : " << Book2.subject <<endl;

    cout << "Book 2 id : " << Book2.book\_id <<endl;

    return 0;

}

**Output**

**Kesimpulan**

Pada program tersebut menggunakan fungsi struct untuk menyimpan data. Terdapat variabel Books yang berada dalam fungsi struct, kemudian pada fungsi tersebut terdapatvariabel book1 dan book2. Untuk memanggilnya dapat menggunakan book1.element dan book2.element.

* **Tugas 2**

**Source Code**

#include <iostream>

#include <string.h>

using namespace std;

void printBook(struct Books book);

struct Books{

    char title[50];

    char author[50];

    char subject[100];

    int book\_id;

};

int main( ){

    system("cls");

    struct Books Book1; *// Declare Book1 of type Book*

    struct Books Book2; *// Declare Book2 of type Book*

*// book 1 specification*

    strcpy(Book1.title, "Learn C++ Programming" );

    strcpy(Book1.author, "Chand Miyan" );

    strcpy(Book1.subject, "C++ Programming" );

    Book1.book\_id = 6495407;

*// book 2 specification*

    strcpy(Book2.title, "Telecom Billing" );

    strcpy(Book2.author, "Yakit Singha");

    strcpy(Book2.subject, "Telecom");

    Book2. book\_id = 6495700;

*// Print Book1 info*

    printBook(Book1);

*// Print Book2 info*

    printBook(Book2);

    return 0;

}

void printBook(struct Books book){

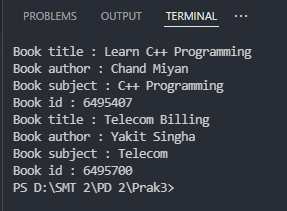
    cout << "Book title : " << book.title <<endl;

    cout << "Book author : " << book.author <<endl;

    cout << "Book subject : " << book.subject <<endl;

    cout << "Book id : " << book.book\_id <<endl;

}

**Output**

**Kesimpulan**

Pada program tersebut mencoba suatu struktur ke fungsi yang menerima input struktur dengan element yang sama. Karena fungsi yang baru menerima variabel struktur dengan element yang sama dengan variabel struktur yang di passing, maka element akan otomatis terisi ke variabel struct fungsi yang baru.

* **Tugas 3**

**Source Code**

#include <iostream>

#include <string.h>

using namespace std;

void printBook(struct Books \*book);

struct Books{

    char title[50];

    char author[50];

    char subject[100];

    int book\_id;

};

int main( ){

    system("cls");

    struct Books Book1; *// Declare Book1 of type Book*

    struct Books Book2; *// Declare Book2 of type Book*

*// Book 1 specification*

    strcpy(Book1.title, "Learn C++ Programming" );

    strcpy(Book1.author, "Chand Miyan" );

    strcpy(Book1.subject, "C++ Programming" );

    Book1.book\_id = 6495407;

*// Book 2 specification*

    strcpy(Book2.title, "Telecom Billing" );

    strcpy(Book2.author, "Yakit Singha");

    strcpy(Book2.subject, "Telecom");

    Book2.book\_id = 6495700;

*// Print Book1 info, passing address of structure*

    printBook(&Book1);

*// Print Book1 info, passing address of structure*

    printBook(&Book2);

    return 0;

}

*// This function accept pointer to structure as parameter.*

void printBook(struct Books \*book){

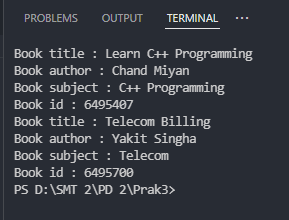
    cout << "Book title : " << book->title <<endl;

    cout << "Book author : " << book->author <<endl;

    cout << "Book subject : " << book->subject <<endl;

    cout << "Book id : " << book->book\_id <<endl;

}

**Output**

**Kesimpulan**

Pada program tersebut cara kerjanya hampir sama dengan Tugas 3, namun untuk fungsi yang baru menerima input berupa pointer. Berdasarkan teori untuk memanggil variabel pointer struct dapat menggunakan variabel\_name -> element.

* **Tugas 4**

**Source Code**

#include <string.h>

#include <iostream>

#include <stdio.h>

using namespace std;

struct mahasiswa{

char nama[50];

int no\_id;

char gender[5];

char kelas[10];

};

int main(){

    system("cls");

    struct mahasiswa mhs[100];

    int i,jml\_mhs;

    printf("Masukkan jumlah mahasiswa : ");

    scanf("%d",&jml\_mhs);

*// ===== Memasukkan data-data mahasiswa ===== //*

    for(i=1;i<=jml\_mhs;i++){

        printf("Mahasiswa ke-%d : \n",i);

        printf("Nama : "); scanf("%s",mhs[i].nama);

        printf("No Id : "); scanf("%d",&mhs[i].no\_id);

        printf("Gender : "); scanf("%s",mhs[i].gender);

        printf("Kelas : "); scanf("%s",mhs[i].kelas);

        printf("\n\n");

    }

*// ===== Mencetak data-data mahasiswa ===== //*

    for(i=1;i<=jml\_mhs;i++){

        printf("Mahasiswa ke-%d : \n",i);

        printf("Nama : %s\n",mhs[i].nama);

        printf("No Id : %d\n",mhs[i].no\_id);

        printf("Gender : %s\n",mhs[i].gender);

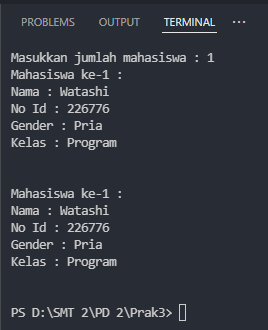
        printf("Kelas : %s\n",mhs[i].kelas);

        printf("\n\n");

    }

    return 0;

}

**Output**

**Kesimpulan**

Pada program tersebut menggunakan array of struct. Jadi struktur dipakai seperti array sehingga kita memiliki variabel struktur tersebut dengan banyak atau dengan kata lain kita memiliki tabel yang berisi element yang sama dengan jumlah yang banyak. Penggunaan array of struct sangat berguna untuk menghemat penggunaan variabel.

* **Tugas 5**

**Source Code**

#include <string.h>

#include <iostream>

#include <stdio.h>

using namespace std;

struct anatomi{

    char media\_gerak[50];

    int jml;

};

struct pola\_hidup{

    char cara\_bergerak[50];

    struct anatomi gerak;

    char cara\_berkembangbiak[50];

};

struct jenis{

    char nama[50];

    struct pola\_hidup ph;

};

int main(){

    system("cls");

    struct jenis makhluk[100];

    int i,jml;

    printf("Masukkan jumlah makhluk : ");

    scanf("%d",&jml);

*// ===== Memasukkan data-data makhluk ===== //*

    for(i=1;i<=jml;i++){

        printf("Masukkan nama makhluk : "); scanf("%s",makhluk[i].nama);

        printf("Masukkan cara bergeraknya : ");

        scanf("%s",makhluk[i].ph.cara\_bergerak);

        printf("Dengan media apa makhluk tersebut bergerak : ");

        scanf("%s",makhluk[i].ph.gerak.media\_gerak);

        printf("Berapa jumlah kaki/sirip/sayap makhluk tersebut : ");

        scanf("%d",&makhluk[i].ph.gerak.jml);

        printf("Dengan cara apa makhluk berkembangbiak : ");

        scanf("%s",makhluk[i].ph.cara\_berkembangbiak);

        printf("\n");

    }

    printf("\n\n");

*// ===== Mencetak data-data makhluk ===== //*

    for(i=1;i<=jml;i++){

        printf("Nama makhluk yaitu %s\n",makhluk[i].nama);

        printf("Cara bergerak dengan %s\n",makhluk[i].ph.cara\_bergerak);

        printf("Media makhluk tersebut bergerak menggunakan %s\n",makhluk[i].ph.gerak.media\_gerak);

        printf("Jumlah kaki/sirip/sayap makhluk tersebut adalah %d\n",makhluk[i].ph.gerak.jml);

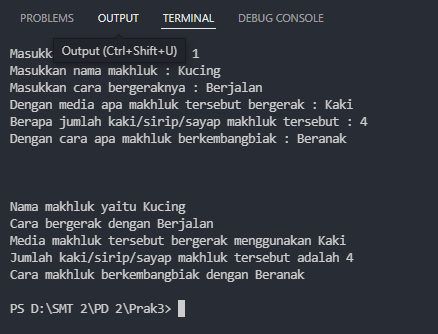
        printf("Cara makhluk berkembangbiak dengan %s\n",makhluk[i].ph.cara\_berkembangbiak);

        printf("\n");

    }

    return 0;

}

**Output**

**Kesimpulan**

Pada program tersebut menggunakan nested struct. Jadi di dalam suatu variabel struct terdapat variabel struct yang lain. Penggunaan nested struct cukup berguna jika element – element dalam satu variabel struct terlalu banyak, sehingga kita bisa memisahkan element – element tersebut untuk dijadikan variabel struct yang lain.